



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/040,149	01/02/2002	Bob Janssen	DVME-1018US	9408
21302 7590 07/03/2007 KNOBLE, YOSHIDA & DUNLEAVY EIGHT PENN CENTER SUITE 1350, 1628 JOHN F KENNEDY BLVD PHILADELPHIA, PA 19103			EXAMINER SCUDERI, PHILIP S	
			ART UNIT 2153	PAPER NUMBER
			MAIL DATE 07/03/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

**Office Action Summary**

Application No.

10/040,149

Applicant(s)

JANSSEN ET AL.

Examiner

Philip S. Scuderi

Art Unit

2153

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 10 June 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-10, 18 and 19 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-10, 18 and 19 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

## DETAILED ACTION

### *Response to Arguments*

#### I. Frese (U.S. Patent No. 5,909,545)

Applicant's arguments filed 10 June 2007 (herein "Remarks") in regards to the Frese reference have been fully considered but they are not persuasive.

1. In the last office action the examiner interpreted the limitation "configured to enable the server ... to control the display on a screen of the display device ... of a screen area having contents generated locally on the client computer" (emphasis added) to merely require the system to be capable of enabling the server to control the display as claimed. The examiner found that Frese's system is capable of enabling server (20) to control the display of a screen area having contents generated locally on client (16) by virtue of the fact that the client (16) and the server (20) are connected via a network (see Frese at fig. 1).

Applicant argues that "merely connecting a client computer (16) to a server (20) is not enough to enable the server (20) to control the display ... as the Examiner suggests" (see Remarks at page 5). Applicant further argues that the server must be provided with means for controlling the display to be capable of controlling the display (see Remarks at page 6). These arguments are not deemed persuasive.

Even if the server needs to be provided with means for controlling the display to be "configured to" control (i.e., capable of controlling) the display, this argument is inconsistent with the language of the claims. The server does not actually need to be capable of controlling the display.

The claims specify that the system is “configured to enable” the server to control the display. Thus, the system merely needs to be capable of enabling the server to control the display. Whether the server is actually provided with means for controlling the display is therefore irrelevant. To be capable of enabling the server to control a display at the client the server needs to be: (1) capable of being provided with means for controlling the display of the client; and (2) connected to the client via some communication network. Almost any computer in existence is capable of being provided with software for controlling the display of a client to which it is connected via a network. The server (20) disclosed by Frese is a conventional server with various applications installed thereon and thus is no exception.

2. In the last office action the examiner took the position that Frese teaches that the server (20) controls the display (specifies RDM applet 18’s parameters) on a screen of the display device (client 16’s display device) of a screen area having contents generated locally on the client computer (RDM applet’s display). Applicant alleges that this position is inaccurate for various reasons (see Remarks at page 6).

2-1. Applicant argues that the RDM applet 18 controls the display of a screen area and not the server. The examiner does not see how this argument is consistent with the language of the claims. The claims recite that the system is configured to enable the server to control the display. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See In re Van Geuns, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

2-2. Applicant argues that RDM executes on the user system 16 and thus the user system 16 controls the display by executing the RDM. The examiner agrees with this statement, however just because the user system 16 controls some aspects of the RDM applet does not mean that the server

Art Unit: 2153

20 does not control any aspects of the applet. The server 20 is an HTTP server that transmits HTML documents to the client 16 (see Frese at col. 7, ll. 28-45). The HTML documents (stored at server 20) comprise applet tags that specify parameters such as platform and colordepth, which are passed to the applet (see Frese at col. 7, ll. 46-55). The applet parameters are used to launch an RDM applet that best corresponds to the user's operating environment (see Frese at col. 7, ll. 64-65; col. 8, ll. 30-33; col. 10, ll. 9-14). Thus, by specifying the applet parameters, the server controls the operating environment displayed by the RDM applet.

2-3. Applicant argues that Frese does not appear to disclose that the server specifies RDM applet 18's parameters. The examiner disagrees. The server is an HTTP server that transmits HTML documents comprising applet tags to the client (see Frese at col. 7, ll. 28-55).

2-4. Applicant argues that the RDM applet does not generate content. The examiner disagrees. The RDM applet generates various content such as "an application window for RDM 18" (see Frese at col. 9, ll. 67 – col. 10, ll. 1).

2-5. Applicant argues that even if RDM applet 18 displays content generated locally, this process does not involve the server 20 and thus the display of the locally generated content is not controlled by the server. The examiner disagrees. Again, the server controls the RDM applet by specifying the parameters that select an RDM applet according to the user's operating environment (see Frese at col. 7, ll. 64-65; col. 8, ll. 30-33; col. 10, ll. 9-14). Thus, any content displayed by the RDM 18 can be considered "controlled" by the server.

3. Applicant argues that Frese does not meet all the limitations of claim 19 because a web page is allegedly not a user interface (see Remarks at page 7). Applicant asserts that Frese's browser 30 is the user interface and the web page is the content (see id.) The examiner disagrees.

The specification does not define a user interface and does not suggest that the term is necessarily limited to particular type of user interface. Thus, the examiner interprets a user interface to broadly mean anything that provides a user with information. A web page clearly provides a user with information and is therefore a user interface as claimed.

Moreover, it was common in the art to refer to a web page or components thereof as a user interface. For example, the article “Designing Web-Based User Interfaces” by Scott Ambler (hereinafter Ambler) states, “the design of a user interface, in this case the design of a form, can clearly be critical to your success” (see Ambler at page 1 (emphasis added)). It is clear from the above quotation that Ambler is referring portions of the actual web page (forms) as “user interfaces.”

4. Applicant argues that the display properties in Frese are specified by browser 30 at the client, rather than by server 20 because the applet parameters and applet tags are allegedly generated by the browser 30 (see Remarks at page 7). The examiner disagrees.

Frese states “[w]hen a user activates the applet tag field, browser 30 transmits a request for activation of the selected application program along with attributes and parameters describing the operating system environment of system 16” (see Frese at col. 7, ll. 41-45 (emphasis added)).

However, this does not mean that the browser 30 actually fills in the parameter fields. One of ordinary skill in the art would readily recognize that servers generate “APP” or “APPLET” tags. In Frese, the user appears to select an applet to display based on the user’s environment.

Perhaps it is possible (albeit unlikely) that the web page taught by Frese dynamically generates the applet tags using Javascript or VBScript (even though the examiner knows of no script that can actually perform these functions it is possible that such script exists). However, even if this

were the case, it would be the server that provides the Javascript or VBScript. Thus, the reference would still meet the limitation that is allegedly in question because providing script that determines the application parameters is “controlling” the display.

## **II. Willems (U.S. Patent No. 5,613,090)**

Applicant's arguments filed 10 June 2007 (herein “Remarks”) in regards to the Willems reference have been fully considered but they are not persuasive.

1. In the pre-appeal conference request filed on 18 January 2007 applicant argued that Willems teaches away from modifying the invention shown in figure 9 by making the window manager remote as described by Willems in regards to prior art figure 8 because doing so would allegedly contradict the primary stated goal of Willems’ invention, which is allegedly reduce network traffic.

The examiner agreed that there might be merit to this argument because Willems does appear to indicate that the primary goal of the invention set forth in figure 9 is to reduce network traffic (see Willems at col. 13, ll. 53-58). However, the examiner noted that Willems does not teach away from modifying the prior art figure 8 disclosed by Willems by enabling the window manager to control a client’s locally run applications. Willems suggests that such a modification would be advantageous because it would reduce the amount of front-end code required (see Willems at col. 14, ll. 1-5).

In applicant’s latest response, applicant now argues that the examiner is again incorrect and that modifying the prior art figure 8 in view of the teachings of figure 9 would not have been obvious for the same reasons that led to withdrawal of the rejection over figure 9 in view of figure 8. Namely, applicant argues that modifying “the embodiment of Fig. 8, wherein the X Windows

Art Unit: 2153

window manager 100 is run remotely on the server, to enable the X Windows window manager 100 to control locally run applications ... will increase network traffic since the window manager 100 of Fig. 8 running on the server will have to send commands over the network.” (see Remarks at page 8). Applicant concludes that this modification would contradict Willems’ primary stated goal of reducing network traffic (see Remarks at page 8). This latest argument is not deemed persuasive.

The main flaw in this latest argument is that the primary stated goal of reducing network traffic that applicant is referring to is the primary stated goal of the invention disclosed by Willems in figure 9 (see Willems at col. 13, ll. 53-58). Willems is silent regarding the primary stated goal of the prior art embodiment of figure 8, which the rejection is based upon. Nowhere does Willems teach or fairly suggest that the primary design goal of prior art figure 8 is to reduce network traffic. To the contrary, Willems suggests that the primary purpose of the invention as set forth in figure 9 is to overcome the fact that the prior art embodiment of figure 8 does not sufficiently reduce network traffic (see Willems at col. 13, ll. 53-58).

For the reasons set forth above, Willems clearly does not teach away from modifying the prior art figure 8 disclosed by Willems by enabling the window manager to control a client’s locally run applications. Indeed, Willems discloses advantages to such a modification such as reducing the amount of front-end code required (see Willems at col. 14, ll. 1-5). Thus, the examiner maintains that it would have been obvious to one of ordinary skill in the art to make this modification.

2. Applicant argues that it would not have been obvious to provide means on the client computer for locally running at least one further application because doing so would allegedly increase network traffic (see Remarks at pages 8-9). The examiner disagrees.

Art Unit: 2153

As detailed above, the primary stated goal of reducing network traffic is the primary goal of the invention of figure 9, not of the prior art embodiment of figure 8 (see Willems at col. 13, ll. 53-58). Thus, even if running at least one further application would increase network traffic, doing so would not contradict the prior art embodiment of figure 8.

3. Applicant argues that Willems' figure 8 depicts window manager 100 and X server 102 as separate elements and Willems states that running the window manager 100 remotely increases network traffic between window manager 100 and X server 102 (see Remarks at page 9). Applicant concludes that Willems does not teach that the user interface is being "run" by X server 102.

The claims specify that the server provides the user interface and controls applications through the user interface. The claims do not specify that the user interface be "run" by the server. Thus, applicant's conclusion that Willems allegedly does not teach that the user interface is being "run" by the server is largely irrelevant because although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See In re Van Geuns, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

The examiner's best understanding is that applicant meant to argue that the X server 102 does not "provide" the window manager. But, Willems does teach this feature as claimed. The specification does not define "providing" a user interface. The examiner therefore interprets the term to broadly mean to make a user interface or any portion thereof available. This is a reasonable interpretation because, for example, MSN Encarta defines provide as "to give, sell, or make available something that is wanted or needed by somebody or something" (see the enclosed MSN Encarta definition). Willems teaches that the window manager must interact with the X server (see Willems

Art Unit: 2153

at col. 13, ll. 45-58). The X server therefore makes the window manager available, which clearly falls within the broadest reasonable interpretation of providing the window manager.

Moreover, Willems does not teach away from running the window manager 100 on the same computer as the X server 102. Indeed, the suggestion that a performance hit is taken when these elements are run on separate computers provides more than adequate motivation to establish that it would have been obvious to combine these elements onto a single computer, the motivation being to avoid to avoid the performance hit (see Willems at col. 13, ll. 53-58).

4. Applicant argues that the prior art does not teach that the display on the client computer is generated according to display properties specified by the server (see Remarks at page 11). The examiner disagrees.

The examiner's position is that it would have been obvious to one of ordinary skill in the art to enable the remote window manager 100 described in reference to figure 8 to control the client's locally run applications to reduce the amount of front-end code required (see Willems at col. 14, ll. 1-5). Willems teaches that the window manager 100 "controls the look and feel" of applications that it manages (see Willems at col. 13, ll. 41-44). So, because the window manager manages locally run applications, it also controls the look and feel (i.e., the display properties) of the locally run applications. And, because it would have been obvious to one of ordinary skill in the art to run the window manager 100 and the X server 102 on the same computer (as detailed above), the server (computer running the X server and the window manager) would clearly specify the look and feel (i.e., the display properties) of the locally run applications.

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

**Claims 1-10, 18, and 19 are rejected under 35 U.S.C. 102(b) as being anticipated by Frese (U.S. Patent No. 5,909,545).**

As to claim 1, Frese teaches a server-based computing system, comprising at least one server (20) and at least one client computer (16), connected to the server (20) through a network, wherein the server (20) comprises means for providing the client computer (16) with a user interface (web page), wherein the client computer (16) comprises an input device for providing input to an application (22) through the user interface (web page) and a display device for presenting output from an application (22) through the user interface (web page), wherein the server (20) comprises means for running the application (22), wherein the client computer (16) comprises means for locally running at least one further application (18), wherein the system comprises means for controlling the locally run applications (18) through the user interface (web page) provided by the server (20). (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

The claims specify that the system is “configured to enable” the server to control the display. Thus, the system merely needs to be capable of enabling the server to control the display. Whether the server is actually provided with means for controlling the display is therefore irrelevant. To be capable of enabling the server to control a display at the client the server needs to be: (1) capable of being provided with means for controlling the display of the client; and (2) connected to the client via some communication network. Almost any computer in existence is capable of being provided with software for controlling the display of a client to which it is connected via a network. The

server (20) disclosed by Frese is a conventional server with various applications installed thereon and thus is no exception.

Moreover, Frese would anticipate the claim even if it actually required the server to control the display on a screen of the display device of a screen area having contents generated locally on the client computer. Specifically, Frese teaches a server (20) that controls the display on a screen of the display device of a screen area (specifies RDM applet 18's parameters) having contents generated locally (RDM applet 18's display) on the client computer (16) (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 2, Frese teaches means for controlling an application (22) running on the server (20) and further applications (18), running locally, through the user interface (web page) (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 3, Frese teaches that the user interface (web page) comprises means for initiating a locally run application (18) (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 4, Frese further teaches means for presenting an overview of available applications installed on the server and on the client computer through the user interface (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 5, Frese teaches that the user interface comprises means for presenting an overview of applications running on the server (20) (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 6, Frese teaches that the user interface (web page) comprises means for generating a merged local client screen for display on the display device (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 7, Frese teaches that the server (20) comprises means for controlling the display of the merged local client screen on the display device (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 8, Frese teaches that the client computer (16) comprises means for generating a local client screen area (browser), comprising visual output from the locally run applications (18), and the server (20) comprises means for generating a screen area, wherein the system comprises means for merging the local client screen area and the screen area generated by the server, to form a local client screen (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 9, Frese teaches means for automatically updating the local client screen, when changes occur in the local client screen or in the screen area generated by the server (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 10, Frese teaches means for selecting a running application; and means for providing input to the selected application or means for presenting output from the selected application on the client computer (16) through the user interface (web page) (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 18, Frese teaches a computer program stored on a computer readable medium, wherein the computer program can be loaded onto a server (20) through a network to a client computer (16) wherein the client computer (16) comprises an input device for providing an input to an application (22) through a user interface (web page) and a display device for presenting output from an application (22) through the user interface (web page), wherein the server (20) comprises means for running the application (22), wherein the client computer (16) comprises means for locally running at least one further application (18), and wherein the server (20) and client computer (16) comprise means for controlling the locally run applications (18) through a user interface (web

Art Unit: 2153

page) provided by the server (20), wherein the computer program, when run on the server (20) instructs the server (20) to provide the client computer (16) with the user interface (web page) and the server (20) controls (by specifying applet parameters) the display on a screen of the display device having contents generated locally on the client computer (16) (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

As to claim 19, Frese teaches a computer program stored on a computer readable medium, wherein the computer program can be loaded onto a computer (16), the computer (16) being connected through a network to a server (20), and comprising an input device for providing input to an application (22) through a user interface (web page) and a display device for presenting output from an application (22) through the user interface (web page), wherein the server (20), comprises means for running an application (22), and wherein the computer comprises means for locally running at least one further application (18), wherein the computer program, when run on the computer (16), causes the computer to accept the user interface (web page), the user interface (web page) being configured for controlling the at least one locally run application (18) and being provided by the server (20), and further causes the computer (16) to display a screen area having contents generated locally (RDM applet 18's display) on the client computer (16) according to display properties (applet parameters) specified by (in an APP tag) the server (20) (see Frese at fig. 1; col. 6, ll. 39 to col. 8, ll. 50).

### ***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the

Art Unit: 2153

subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

**Claims 1-10, 18, and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Willems (U.S. Patent No. 5,613,090).**

As to claim 1, in regards to figure 8, Willems describes a prior art X Windows environment, comprising at least one server (X server) and at least one client computer, connected to the server (X server) through a network, wherein the client computer comprises an input device for providing input to an application (X applications) through a user interface (remote window manager 100) and a display device for presenting output from an application (X applications) through the user interface (remote window manager 100), wherein the server (X server) comprises means for running the application (X applications) (see Willems at fig. 8; col. 13, ll. 39-58).

Willems does not expressly disclose that the server (X server 102) comprises means for providing the client computer with a user interface (remote window manager 100). However, Willems does not teach away from running the window manager 100 on the same computer as the X server. Willems discloses there is a performance hit taken when the server (X server 102) and the means for providing the user interface (remote window manager 100) are on separate machines (see Willems at col. 13, ll. 53-58). It would have been obvious to one of ordinary skill in the art to install the X server 102 and the remote window manager 100 on the same computer to avoid this performance hit (see id.)

Willems does not expressly disclose that the client computer comprises means for locally running at least one further application. Nonetheless, it was well known in the art to provide a client computer with means for running local applications. It would have been obvious to provide such a means here so that users can run their own local applications.

The prior art shown in figure 8 of Willems does not teach controlling locally run applications through the user interface (window manager 100) provided by the server (X server). However, enabling a window manager (i.e., a user interface) to control locally run applications was well known in the art, as evidenced by Willems discussion in regards to figure 9.

In regards to figure 9, Willems describes a window manager that controls a client's locally run applications and provides advantages such as reducing the amount of front-end code required. See Willems at fig. 9; col. 13, ll. 59 to col. 14, ll. 13. It would have been obvious to modify the prior art window manager of figure 8 by enabling it to control a client's locally run applications to reduce the amount of front-end code required (see Willems at col. 14, ll. 1-5).

Claim 1 recites "the client computer ... is configured to enable the server ... to control the display on a screen of the display device ... of a screen area having contents generated locally on the client computer" (emphasis added). To meet this limitation Willems' client computer merely needs to be capable of enabling the server (X server) to control the display on a screen of the display device of a screen area having contents generated locally on the client computer (16) (see MPEP § 2111.04). Willems' client computer is connected to the server (X Server) via a network and is therefore capable of enabling the server (X Server) to control the display on a screen of the display device of a screen area having contents generated locally on the client computer (see Willems at fig. 9).

Moreover, Willems would obviate the claims even if they actually required the server to control the display on a screen of the display device of a screen area having contents generated locally on the client computer because enabling the window manager to control a client's locally run applications would enable it to control the display on a screen area of the display device of a screen

Art Unit: 2153

area having contents generated locally on the client computer (see Willems at fig. 9; col. 13, ll. 59 to col. 14, ll. 13).

As to claim 2, Willems further teaches means for controlling an application running on the server and further applications, running locally, through the user interface (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13).

As to claim 3, Willems does not expressly disclose that the user interface comprises means for initiating a locally run application. However, it was well known in the art that X Windows environments provide means for initiating locally run applications. It would have been obvious to provide means for initiating locally run applications here to enable users to conveniently initiate the locally run applications (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13).

As to claims 4 and 5, Willems does not expressly disclose presenting an overview of available applications installed on the client and/or the server. However, it was well known in the art that X Windows environments provide means for presenting an overview of available applications. It would have been obvious to provide means for presenting an overview of available applications to enable users to conveniently locate available applications.

As to claim 6, Willems further teaches means for generating a merged local client screen, for display on the display device (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13)

As to claim 7, Willems further teaches that the server comprises means for controlling the display of the merged local client screen on the display device (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13).

As to claim 8, Willems further teaches that the client computer comprises means for generating a local client screen area, comprising visual output from the locally run applications, and the server comprises means for generating a screen area, wherein the system comprises means for

Art Unit: 2153

merging the local client screen area and the screen area generated by the server, to form the local client screen area (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13).

As to claim 9, Willems further teaches means for automatically updating the local client screen, when changes occur in the local client screen, when changes occur in the local client screen area and/or in the screen area generated by the server (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13).

As to claim 10, Willems further teaches means for selecting a running application; and means for providing input to the selected application and/or means for presenting output from the selected application on the client computer through the user interface (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13).

As to claim 18, in regards to figure 8, Willems describes a prior art X Windows environment, comprising a computer program stored on a computer readable medium, wherein the computer program can be loaded onto a server (X server) connected through a network to a client computer wherein the client computer comprises an input device for providing input to an application (X applications) through a user interface (window manager 100), wherein the server (X server) comprises means for running the application (X applications) (see Willems at fig. 8, 9; col. 13, ll. 39 to col. 14, ll. 13).

Willems does not expressly disclose that the server (X server 102) provides the client computer with the user interface (remote window manager 100). However, Willems does not teach away from running the window manager 100 on the same computer as the X server. Willems discloses there is a performance hit taken when the server (X server 102) and the means for providing the user interface (remote window manager 100) are on separate machines (see Willems at col. 13, ll. 53-58). It would have been obvious to one of ordinary skill in the art to install the X

Art Unit: 2153

server 102 and the remote window manager 100 on the same computer to avoid this performance hit (see *id.*)

Willems does not expressly disclose that the client computer comprises means for locally running at least one further application. Nonetheless, it was well known in the art to provide a client computer with means for running local applications. It would have been obvious to provide such a means here so that users can run their own local applications.

The prior art shown in figure 8 of Willems does not teach controlling locally run applications through the user interface (window manager 100) provided by the server (X server). However, enabling a window manager (i.e., a user interface) to control locally run applications was well known in the art, as evidenced by Willems discussion in regards to figure 9.

In regards to figure 9, Willems describes a window manager that controls a client's locally run applications and provides advantages such as reducing the amount of front-end code required. (see Willems at fig. 9; col. 13, ll. 59 to col. 14, ll. 13). It would have been obvious to modify the prior art window manager of figure 8 by enabling it to control a client's locally run applications to reduce the amount of front-end code required (see Willems at col. 14, ll. 1-5).

Willems obviate the limitation "the server controls the display on a screen of the display device of a screen area having contents generated locally on the client computer" because enabling the window manager to control a client's locally run applications would enable it to control the display on a screen area of the display device of a screen area having contents generated locally on the client computer (see Willems at fig. 9; col. 13, ll. 59 to col. 14, ll. 13).

As to claim 19, Willems teaches a computer program stored on a computer readable medium, wherein the computer program can be loaded onto a computer, the computer being connected through a network to a server (X server), and comprising an input device for providing

Art Unit: 2153

input to an application (X applications) through a user interface (window manager 100) and a display device for presenting output from an application (X applications) through the user interface (window manager 100), wherein the server (X server), comprises means for running an application (X applications). (see Willems at fig. 8; col. 13, ll. 39-58).

Willems does not expressly disclose that the server (X server 102) provides the client computer with the user interface (remote window manager 100). However, Willems does not teach away from running the window manager 100 on the same computer as the X server. Willems discloses there is a performance hit taken when the server (X server 102) and the means for providing the user interface (remote window manager 100) are on separate machines (see Willems at col. 13, ll. 53-58). It would have been obvious to one of ordinary skill in the art to install the X server 102 and the remote window manager 100 on the same computer to avoid this performance hit (see id.)

Willems does not expressly disclose that the computer comprises means for locally running at least one further application. Nonetheless, it was well known in the art to provide a client computer with means for running local applications. It would have been obvious to provide such a means here so that users can run their own local applications.

The prior art shown in figure 8 of Willems does not teach that the user interface (window manager 100) controls the at least one locally run application and causes the computer to display a screen area having contents generated locally on the client computer according to display properties specified by the server. However, enabling a window manager (i.e., a user interface) to control locally run applications was well known in the art, as evidenced by Willems discussion in regards to figure 9.

Art Unit: 2153

In regards to figure 9, Willems describes a window manager that controls a client's locally run applications and provides advantages such as reducing the amount of front-end code required. (see Willems at fig. 9; col. 13, ll. 59 to col. 14, ll. 13). It would have been obvious to modify the prior art window manager of figure 8 by enabling it to control a client's locally run applications to reduce the amount of front-end code required. (see Willems at col. 14, ll. 1-5). Enabling the window manager of figure 8 to control the client's locally run applications would cause the user interface to display a screen area having contents generated locally on the client computer according to display properties specified by the server.

### ***Conclusion***

**THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.


Any inquiry concerning this communication or earlier communications from the examiner should be directed to Philip S. Scuderi whose telephone number is (571) 272-5865. The examiner can normally be reached on Monday-Friday 9:00 am - 5:30 pm.

Art Unit: 2153

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glenton B. Burgess can be reached on (571) 272-3949. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

PS



GLENTON B. BURGESS  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100